

MEMORIES FOR ELECTRONIC SYSTEMS

TECHNICAL FIELD OF THE INVENTION

[0001] The present invention relates in general to electronic systems in which memories are used for data storage, as well as program storage. It also relates to uniprocessor and multiprocessor systems, in computer, communication and consumer markets. In particular is described a memory architecture for fixed, as well as variable packet lengths.

BACKGROUND OF INVENTION

[0002] With the advent of uniprocessor personal computers, multiprocessor server systems, home networking, communications systems, routers, hubs, switch fabrics, cell phones, PDA's, and mass storage servers, technology has rapidly advanced in order to support the exchange of digital data between these and similar devices. To this end, new protocols have been developed to adapt to the use of the digital data format, instead of the older analog data format. Standards in communications between the ever increasing number of different devices capable of digital data transmission and reception, are evolving. Communication between a telecommunications base station and a cell phone is a primary example. Another example is PC-centered home network communicating with numerous electronic appliances. The implementation of these standard protocols is a nontrivial problem which must be addressed at all levels during both software and hardware development. Moreover, mobile systems, like cell phones, require lower operating power with added features and performance.

[0003] As one example, the Integrated Services Digital Networks (ISDN) protocol is one particular format which has been adopted to support digital data telecommunications from various sources, including digital telephones and faxes,

personal computers and workstations, video teleconferencing systems, and others. Extensions of the ISDN protocol include the broadband ISDN (BISDN) protocols which support the exchange of large files of data and/or data with strict time restrictions, such as full motion video. One of these broadband ISDN protocols, the Asynchronous Transfer Mode (ATM) protocol, is being broadly accepted in the telecommunications industry. Other protocols, like Internet Protocol, are also very popular, especially since voice over IP is rapidly gaining acceptance. IP, Ipv6, TCP, UDP, MPLS, UMTS, GPRS, CDMA, GSM, Ethernet, WAP, H.323, MGCP, SIP, RTP, Frame Relay, PPP, SS7, X25 are some other protocols beyond ATM.

[0004] Data formats for the data packets of the various different protocols vary greatly. Broadly, they can be described as: 1) Perfectly-sized packets and 2) imperfectly-sized packets. Perfectly-sized packets are octal multiples - namely those comprised of 16, 32, 64 or 128 bytes. These find applications in computing and communications memories, and hence, memory device data architectures - stand alone or embedded - which adhere to x 4, x 8, x 16, x 32 or x 9, x 18 data formats (with parity). Perfect-sized packets optimize bandwidth from the memory.

[0005] Imperfectly-sized packets are those which 1) utilize non-octal multiples, and, 2) utilize a data format which does not adhere to a length = 2^n bits, where n is an even number. For example, in some Ethernet applications, data packet size can be 20 bytes. Another example is packet-over-sonet where minimum data transfer size is 40 bytes. Hence, with traditional RAM's, one will incur a bandwidth inefficiency while reading and writing out of such memory devices.

[0006] In order to adhere to one protocol, and maximize bandwidth and latency, a memory core has to be organized architecturally, in one manner. In order to accommodate several protocols in the same memory, where bandwidth and latency are optimized, the memory core architecture has to be different.

[0007] In essence, the ATM protocol implements time-division concentration and packet switching to connect two or more end users through a public or private switched

network, including the routers, switches and transmission media. Generally, streams of data are divided into time slots (cells) which are made available on demand (in contrast to the Synchronous Transfer Mode where each slot is preassigned). In ATM, a standard cell is 53 bytes long, with the first 5 bytes being the header and the following 48 bytes containing user data. The number of data bytes in nonstandard cells can be as small as few bytes, to as large as 4Kbytes, depending on the protocol used. The header includes fields for flow control data and management, virtual path and virtual channel identifiers, and a payload identifier, and generally defines the packet switching of each cell. The user data bytes contain the user data itself, along with an adaptation layer (header and trailer) which identifies the data type, data length, data starting and ending bytes, etc.

[0008] There are several means of packet switching used in protocol-based systems. One method uses shared-memory switches. This shared memory is also called communication system memory in the wired communication industry (routers, servers, network/switch fabrics etc). Here, the user part of each cell is received through a corresponding port and stored in memory. In accordance with a corresponding timing protocol, these data are accessed through a second designated port to complete the switching of the user part of the packet.

[0009] Current shared-memory switches are constructed using static random access memory (SRAM) devices and dynamic random access devices (DRAM). In comparison with dynamic random access memories (DRAMs), SRAMs have a simpler interface, do not require periodic refresh of the data, and are typically faster. However, SRAMs are more expensive, consume much more power, and have lower cell densities. While memory speed remains important in many applications, including those involved with telecommunications, increasing attention must be made to the factors of cost, size and power consumption in order to remain competitive in the marketplace. Hence, a need has arisen for shared-memory switch which has the high performance of an SRAM and the lower cost and reduced power consumption of a DRAM. RLDRAM I/IITM,

FCRAMTM, DDRSDRAM are some of the recent DRAM's that are trying to serve these requirements, with some, but not complete, success. Among other things, all of the above memories utilize a data format that is an even multiple of eight' (or byte oriented) - 8, 16 or 32 (9 or 18 with parity), which does not maximize bandwidth and utilization. In addition, the memory used in portable electronic appliances (e.g., cell phones) used for any communication, are also 'packet data oriented'. To enhance bandwidth at minimum operating power, a need has arisen to optimize memory architecture - although, the transmitting and receiving ports are not many.

SUMMARY OF INVENTION

[0010] The present inventive concepts are embodied in a switch comprising a plurality of ports for exchanging data words of a predetermined word-width, or variable word-width, and a shared-memory for enabling the exchange of data between first and second ones of the ports. The word-width can also be programmed (variable wordwidth) so that multiple protocols can share the same memory through the intervention of a memory controller. In one embodiment, the shared-memory includes an array of memory cells arranged as a plurality of rows, and a single column having a width equal to the predetermined word-width. The shared-memory further includes circuitry for writing a selected data word presented at the first one of the data ports to a selected row in the array during a first time period and for reading the selected data word from the selected row during a second time period to the second one of the ports. The shared memory interfaces to memory controller, which provides the appropriate address, command and control signals. The memory controller can be specific to a particular memory - namely DDRSDRAM, RLDRAM, FCRAM, SRAM, MAGRAM, NVRAM, FeRAM and similar memories (sometimes called universal memory). It can also be an integral part of the overall system controller.

[0011] The inventive concepts are also embodied in a shared-memory switch. A plurality of ports are included for exchanging data between external devices associated

with each of the ports. Each port is also associated with a buffer for assembling a stream of data words being input into the switch into a single word of a predetermined width and for converting single data words of the predetermined width being output from the switch into a stream of data words. The switch includes a shared-memory for effectuating a transfer of data from a first one of the ports to a second one of the ports through corresponding ones of the buffers. The shared-memory comprises a plurality of banks, each having an array of memory cells arranged as a plurality of rows and a single column of the predetermined width and circuitry for selecting a row in response to a received address. A plurality of available address tables each maintain a queue of addresses available for writing the single words of data to a corresponding one of the banks and a plurality of used address tables each maintain a queue of addresses for reading from a corresponding one of the banks.

[0012] A digital information system is also disclosed which includes first and second resources operable to exchange data in a selected digital format and a digital switch. The digital switch has first and second ports for selectively coupling the first and second resources and a shared-memory for enabling the exchange of data between the first and second ports as words of a predetermined word-width. In one embodiment, the shared-memory includes an array of memory cells arranged as a plurality of rows and a single column having a width equal to the predetermined word-width. Additionally, the shared-memory includes circuitry for writing a selected data word presented at the first one of the ports to a selected row in the array during a first time period and for reading the selected data word from the selected row during a second time period to a second one of the ports. In another embodiment, column groups in a given row can be selected randomly, where each of the column groups has a predetermined width from a few bytes up to 4K bytes.

[0013] The present inventive concepts are also embodied in methods for switching a plurality of streams of data, each comprising a selected number of words. A first one of the streams of data is received at a first port to a shared-memory switch

during a first write period. The first stream of data is stored as a first single data word in a first row in shared-memory within the shared-memory switch from which the first single data word is to be subsequently retrieved. A second one of the streams of data is received at a second port to the shared-memory switch during a second write period. Since the shared-memory comprises one or more of random access memories with multibank architectures, it is preferably stored in a next available bank. The memory controller can handle this very easily. In a cyclic memory or "round robin" bank scheme, the next row automatically is stored in the next bank. The second stream of data is stored as a second single data word in a second row in shared-memory from which the second data word is to be subsequently retrieved. The first single data word is retrieved from the first row in shared-memory during a first read period and outputted as the first stream of data through a selected port of the switch. The second data word is retrieved from the second row in shared-memory during a second read period and outputted as the second stream of data through a selected port of the switch. Or, the second row may be read from the same row in the same bank, in one embodiment.

[0014] In another embodiment, the shared memory comprises a plurality of banks, each having an array of memory cells arranged as a plurality of rows and multiple "column groups". The banks also include respective row decoders and "column group" decoders for appropriate access of a given packet. Unlike traditional DRAM's where any single column can be selected, in this invention, one "column group" of any given row can be selected. Within any given row, such groups can vary from 1 to 256, based on realities (manufacturable at a cost the market is willing to accept) of practical integrated circuits. The minimum "group" size can be a few bytes to 1K bytes. Appropriate control circuitry is also included, where column groups can be prefetched, in a sequence or interleave, as is done with "burst length specific double data rate RAMs". A plurality of available address tables each maintain a queue of the addresses available for writing multiple words to corresponding one of the banks and a plurality of

address tables each maintain a queue of addresses for reading from a corresponding one of the banks.

[0015] The present inventive concepts have substantial advantages over the prior art. Most importantly, the present inventive concepts allow for the construction and use of a shared-memory switch which has the high performance of an SRAM and the lower cost of and reduced power consumption of a DRAM. The word DRAM here is not limited to commodity DRAM's only- ferroelectric RAM's or any other read/write memory (universal memory) can also implement these inventive concepts.

BRIEF DESCRIPTION OF DRAWINGS

[0016] For a more complete understanding of the present invention, and the advantages thereof, reference is now made to the following descriptions taken in conjunction with the accompanying drawings, in which.

[0017] FIGURE 1A is a block diagram of a shared-memory switch to which the concepts of the present invention may be advantageously applied;

[0018] FIGURE 1B is a timing diagram, generally describing the operation of the shared-memory of FIGURE 1A;

[0019] FIGURE 2A is a block diagram of a memory suitable for use as the shared-memory in one embodiment of shared-memory switching applications;

[0020] FIGURE 2B is a block diagram of a memory suitable for use as shared memory in another embodiment.

[0021] FIGURE 2C is a block diagram of a memory suitable memory in yet another embodiment.

[0022] FIGURE 3A illustrates the sequence of accesses without compensation cycles for read-write conflicts;

[0023] FIGURE 3B illustrates the sequence of accesses with an additional number of cycles for read/write conflict compensation, some of which are used for refresh operations;

[0024] FIGURE 3C shows an alternate sequence of accesses with an additional number of cycles for read/write conflict compensation;

[0025] FIGURE 4 is a timing diagram of a set of signals defining the DRAM shared-memory interface; and

[0026] FIGURE 5 is a conceptual diagram of a switching system utilizing a shared-memory switch according to the inventive concepts.

[0027] FIGURE 6 shows an embodiment where the READ and WRITE paths for data (in and out of the memory) are separate thus doubling bandwidth and improved bus efficiency.

[0028] FIGURES 7 through 11 illustrate the operation of inventive concepts, where the address, command, control and data, are strobed on both edges of the system clock (rising and falling).

DETAILED DESCRIPTION OF THE INVENTION

[0029] The principles of the present invention and their advantages are best understood by referring to the illustrated embodiment depicted in FIGURES 1-11 of the drawings, in which like numbers designate like parts.

[0030] FIGURE 1A is a block diagram of a shared-memory switch 100 to which the concepts of the present invention may be advantageously applied. A shared-memory may comprise one or more integrated circuits, modules or chassis. In this example, switching is between eight (8) network segments 101, although the actual number of network segments will vary from system to system. The 48-byte user part of each incoming and outgoing packet (the "user data packet") is in an 8 x 48-bit format (i.e., a stream of eight words each forty-eight bits wide). Data is stored however in shared-memory 102 in a 1 x 384-bit format. The requisite conversion between formats is implemented through a corresponding set of buffers 103. The 1 x 384-bit format port for each memory buffer 103 is coupled to a bus 104 which in turn is coupled to a 1 x 384-bit wide port to shared (communication) memory 102. A memory controller - on or off chip - provides CLK, Address, COMMAND and CONTROL features, as well as appropriate data format control. The programmable data format can also be executed through an on-chip mode register, as is done today in RAM'S - examples are "column group selection", burst length mode, burst format (sequence or interleave). Communications memory may be implemented as a double data rate (DDR), quad data rate (QDR), Rambus®, or programmable burst bit length memory to name only a few options.

[0031] The operation of shared-memory 102 can be illustrated in conjunction with the timing diagram of FIGURE 1B. A reference clock CLK provides the time-base, while the address (ADD) and command/control signals like output enable (/OE) and write enable (/WE) allow for data to be written to and read from locations in memory. Each data word is labeled with a designator Rx.y for reads or Wx.y for writes, where x designates the port accessing memory and y designates the word number for the

current word through the corresponding buffer 103. A "P" indicates that a complete user data packet is being transferred.

[0032] In this example, one complete user data packet is exchanged between shared-memory 102 and one of the memory buffers 103 per clock cycle. Specifically, eight writes are made to shared-memory 102 from each memory buffer 103 in sequence, starting with memory buffer 103a (Port 1), as shown in the DATA trace of FIGURE 1B. These writes are followed by eight reads from shared-memory 102 to the memory buffers 103 in sequence, starting with memory buffer 103a (Port 1). In other words, each port is assigned fixed slots for reading and writing to shared-memory. Here, the write latency is zero clock cycles and the read latency is one clock cycle.

[0033] At the same time data is being exchanged between shared-memory 102 and memory buffers 103, data is being exchanged between the ports and memory buffers 103. An exemplary timing of these reads and writes is shown in the bottom eight traces of FIGURE 1B. (For purposes of discussion a single data rate embodiment is assumed.) In this case, eight writes of 48-bit words on 8 consecutive clock cycles (collectively one user data packet) followed by eight reads of 48-bit words on eight consecutive clock cycles are performed to each port. The accesses are staggered from port to port, for example on one clock cycle word 1 of a packet to port 1 is read and word 8 of a packet to port 2 is written, on the next clock cycle word 2 to port 1 is read and word 1 of a packet to port 2 is read and so on. The pattern is the same for all 8 ports.

[0034] In ATM, the data rate is 155.52 Mbit/sec, and therefore 2.72 μ sec are required to transfer a complete ATM cell between a given two ports. In the current example, this is equivalent to 17 clock cycles. This in turn dictates that the shared-memory be accessed every 160 ns (for a switch with 64 ports, this time is reduced to only 20 nsec.). This is an illustrative example only.

[0035] FIGURES 2A, 2B, and 2C are block diagrams of a memory 200 suitable for use as the shared-memory is shared-memory switching applications, such as that

described above. They describe three different embodiments. In the illustrated embodiment of FIG. 2A, memory 200 is constructed of four banks 201. Each bank is based upon a 64k x 384-bit DRAM cell array 202. Preferably, arrays 202 are organized as 64k rows and one column, with each row holding one word of 384-bits (i.e., one entire 48 byte ATM user packet). (In alternate embodiments of the memories shown in FIGURES 2A-2C, the array organizations, and/or the word width of 384-bits, may vary.) The advantage of having rows that are exactly one column (364-bits) wide is the resulting simplified interface to memory 200. Specifically, during an access, only a row address is required, rather than a row address and at least one column address, as is required for array accesses in conventional memory architectures. ATM, Sonet, IP, and Ethernet have varying bit widths from as small as 32 bits (4 bytes) to as large as 4,096 bits- the invention here comprehends all cases. The memory controller or on-chip mode register can effectively program the data path flow, as well as address mapping for multiplexed or nonmultiplexed (broadside) addressing.

[0036] Each bank also includes conventional DRAM sense amplifiers 203, row decoders 204 wordline drivers 205, and control logic 206. Multiplexers 207 allow the 384 cells of an addressed row of a given bank to be accessed through conventional read/write amplifiers 208 under the control of memory control circuitry 209 (i.e., provides for bank switching which will be discussed later). Column decoders, though not shown, can easily be accommodated adjacent to the sense amplifiers.

[0037] Again, DRAM is mentioned generically. The concepts apply to all read/write memories including FCRAM, RLDRAM, nonvolatile memory devices like flash, FeRAM, MAGRAM etc. A page in today's read/write memory can accommodate up to 8,192 bits- the ATM "384 bit" is an example only. Varying bit widths that are protocol dependent as well as interface dependent, are comprehended in this embodiment.

[0038] In the embodiment of FIGURE 2A, four banks 201 are depicted for illustrative purposes. In actual applications, the number of banks is a function of both

the random access latency of each DRAM bank in a given fabrication process and the data required rate of the shared-memory. For example, a high-speed network switch may require a shared-memory access every 10 nsec. Therefore, if the random access latency for each bank is 40 nsec, then four banks are required, (i.e., 40 nsec-10 nsec). Similarly, if the random access latency is 60 nsec, then the number of banks would have to be increased to six. One can also cycle the banks in sequence. DDR (Double data rate) and QDR (Quad data rate) data throughput are also possible extensions of these concepts.

[0039] It should be noted that to achieve lowest latency possible, the dimensions of the memory subarrays composing DRAM arrays 202 must be carefully controlled. For example, a shorter subwordline will allow a selected row to be opened and closed faster than a long row with correspondingly long subwordline. Additionally, shorter bitlines will allow for faster pre-charge and data transfer operations. The exact dimensions (e.g., the number of bits per bitline and number of gates per subwordline) will depend on the process technology selected as well as the required data rate of the system.

[0040] Each bank 201 is associated with an available address table 210. Available address tables 210 are preferably first-in first-out (FIFO) memories which are initialized at either system power up or reset to initially contain all the available addresses to the corresponding bank. For the 64k row arrays of the illustrated embodiment, each available address table 210 maintains a queue of addresses 0 through 65536, (one address associated with each row in the array). During a write to selected bank 201, a bank select or similar signal from a controlling device or system initiates the access and the next available address in the FIFO queue is used to store the data in the corresponding row in the cell array. The address is also copied to the controlling device or system such that the user data packet can be associated with the corresponding header which is being processed in parallel. As additional writes are

made to the array, additional addresses in the corresponding available address table are assigned from the queue.

[0041] As reads from a given bank 201 are performed, the read address is written into the available address table for reuse. The exception to the point to multiple-point data payload switching. In this case, the address of the multipoint payload is not written into the table until the payload has been written to the last port associated with that bank.

[0042] A used address table 211 also is provided for each bank 201. As data is written into each bank, the write address is obtained from the next available address table 210 associated with that bank as described above and input to the appropriate row decoder 204 for the write operation. The write address is also input to the used address table 211 for the bank. Used address tables 211 could be either co-located with the corresponding bank 201 or could be physically located in another part of the switch system. The addresses in the used address tables 211 represent the data. A switch control algorithm can manipulate these addresses by altering their order such that those addresses and correspondingly the associated data can be read out in a selected manner. The used address table is preferably random access memory, and could be either static or dynamic. It should be noted that each of the four used address tables 211 shown in the embodiment of FIGURES 2A- 2C can be managed either independently or as a single memory space.

[0043] It should be recognized that the one system controller (or controllers in a multiple switch system) can directly generate and control the addressing of banks 201. Under this alternative, during writes to a selected bank, the system controller generates the 16-bit address of the location to which the data is to be written. A multiplexer or similar circuitry is used to switch from the corresponding available address table to system controller addressing. The system controller itself stores this address along with the header associated with that data in, for example, system memory by direct addressing. The data can be stored randomly within the assigned bank, or in any

arrangement determined by the controller. During the subsequent read operation, the system controller simply retrieves the appropriate address from memory by direct addressing.

[0044] Exemplary operation of memory 200 in a shared-memory switch can be described in conjunction with TABLE 1. In this case, the switching system uses the four-bank (Banks 1-4) embodiment of memory 200 to support four switching ports (Ports 1-4). Hence, each bank becomes the queue for a corresponding one of the ports from which the data is to be read to effectuate switching. In TABLE 1, each ATM cell is represented symbolically by a numeral "a.b.", where "a" designates the source (write) Port 1-4 and "b" designates the destination (read) Port 1-4, as well as the accessed Bank 1-4. For example, for the designated cell 1.2, data is received through source Port 1, stored in Bank 2, and read from destination Port 2. Similarly, for cell 4.2, data is received through source Port 1, stored in Bank 2, and read from destination Port 2, and so on. The switching sequences, as well as the input and output ports assigned to each cell in TABLE 1 were arbitrarily selected for discussion purposes, in actuality there are a large number of switching sequences and combinations.

TABLE I

ACCESS	BANK 1	BANK 2	BANK 3	BANK 4
WRITE PORT 1		1.2		
WRITE PORT 2				2.4
WRITE PORT 3	3.1			
WRITE PORT 4	4.1			
READ PORT 1	3.1			
READ PORT 2		1.2		
READ PORT 3		NULL		

READ PORT 4				2.4
WRITE PORT 1			1.3	
WRITE PORT 2			2.3	
WRITE PORT 3				3.4
WRITE PORT 4		4.2		
READ PORT 1	4.1			
READ PORT 2		4.2		
READ PORT 3			1.3	
READ PORT 4				3.4

[0045] TABLE II illustrates the case where a four-bank embodiment of memory 200 is used to support a relatively large number of ports (i.e. 32). Here, each bank provides the queue for more than one read port. In this example, the banks are allocated as follows (although there are many other combinations). Bank 1 is the queue for destination ports 1, 5, 9, 13, 17, 21, 25, and 29, Bank 2 for destination ports 2, 6, 10, 14, 18, 22, 26, and 30, Bank 3 for destination ports 3, 7, 11, 15, 19, 23, 27 and 31, and Bank 4 for destination ports 4, 8, 16, 20, 24, 28, and 32. Again, the switching sequences and cell designations for arbitrarily selected.

TABLE II

ACCESS	BANK 1	BANK 2	BANK 3	BANK 4
WRITE PORT 1		1.2		
WRITE PORT 2			2.3	
WRITE PORT 3				3.4
WRITE PORT 4	4.5			

WRITE PORT 5		5.6		
WRITE PORT 6			6.7	
:	:	:	:	:
WRITE PORT 30			30.31	
WRITE PORT 31				31.32
WRITE PORT 32	32.1			
READ PORT 1	32.1			
READ PORT 2		1.2		
READ PORT 3			2.3	
READ PORT 4				3.4
READ PORT 5	4.5			
READ PORT 6		5.6		
.				
.				
READ PORT 31			30.31	
READ PORT 32				31.32

[0046] A

[0047] As discussed above, in the embodiment of FIGURE 2A, each bank 201 has a 64k x 384-bit DRAM array 202. The above tables demonstrate that the memory space of each DRAM array 202 can either be allocated as queue memory for multiple destination ports 101 or could be assigned to the queue of a single destination port 101. Additionally, it is possible that data could be received by given bank 202 faster than it could be transmitted from that bank such that all 64k rows become filled with valid data.

This may result in some rows remaining inaccessible for periods longer than the maximum time allowed between DRAM cell refresh. In such a situation, an extra read operation would be performed to that bank once every 64 memory accesses (i.e., 32 reads followed by 32 writes in a 32-port system). If the accesses are performed at the data rate of ten (10) nsec then an extra read operation is performed to the given bank every 640 nsec. To read 64k rows of data requires approximately 41.9 msec in the present exemplary system. Assuming a DRAM process that requires every row to be refreshed within 64 msec, then by using this technique, no dedicated refresh mode is required for the arrays 202 of memory 200. If however, a situation arises where a refresh is needed, then banks 202 can always be refreshed by simply reading each row in the array in the usual fashion in response to a conventional refresh counter. In this case, 2.6 msec are required to refresh all rows in all four banks.

[0048] The principles of the present invention allow for alternate methods of refreshing data in memory arrays 202. For example, consider again the case where 32 access reads to shared-memory 200 alternate with 32 writes. A series of one four refresh reads is added to thirty-two access reads for the purpose of refreshing 1 to 4 corresponding rows within each array 202. This method adds only a small amount of overhead to the operation of shared-memory 102. For example, if the DRAM process technology requires each row to be refreshed once every 64 milliseconds, then there are 94118 ($94118 = 32 \text{ writes} \times 32 \text{ reads by 4 banks}$) refresh periods which is an adequate number to refresh arrays 202 of the illustrated embodiment.

[0049] FIGURE 2B illustrates an embodiment where "column groups" can be accessed in the same row of a bank. For example, the minimum wordwidth can be 20 bytes (160 bits), instead of 48 bytes (384 bits) for FIG. 2A. There can be 4, 8, 16 or 32 groups of "20 bytes" in a given row. Appropriate address generation by the controller accesses them appropriately. FIG. 2C is another embodiment with different burst lengths of data.

[0050] Conflicts may arise between reads and writes within the same bank. Consider the sequence of accesses illustrated in FIGURE 3A where the letter in each cell designates the operation (read or write) and the number represents the port being accessed.

[0051] Assume for example, that the random access latency for banks 201 is 40 nsecs and that accesses are being performed at a rate of 10 nsecs. Consequently, read operation R1 must be initiated in memory 40 nsec prior to the availability of data at port R1. As a result, in this example, the initiation of the memory operation R1 coincides with the write operation of data from port 29 (W29). Similarly, the R2 read operation begins coincident with the write of data from port 30, the R3 read operation coincident with the write operation W31 and so on. As a result of this timing scheme, write W29 must be to bank 1, write W30 must go to bank 2, write W31 to bank 3 and write W32 to bank 4. Such operating conditions are not allowable since they prevent memory 200 from operating efficiently as a switch. In other words, data is received by a switch randomly without regards to destination. For instance, data being received on write W29 may have a destination port requiring access to banks 2, 3, or 4. These banks, however, are already engaged in reads R30, R31 and R32.

[0052] To remedy the problem of read/write conflicts, eight additional clock cycles (null) are added between the write time slot and the read time slot. At least some of these null time periods could be used for refresh operation as shown in FIGURE 3B. In the case of a thirty-two port system, the total sequence is seventy-two cycles with thirty-two reads, thirty-two writes, four refresh cycles and four null cycles. During Null 1, the read to port 1 (R1) is initiated, during Null 2, read R2 is initiated, at Null 3 read R3 is initiated and at Null 4 read R4 is initiated.

[0053] Another method for controlling the write to read transition is to add only four additional cycles to the thirty-two read-thirty-two write sequence as shown in FIGURE 3C. In this case, read cycle R1 starts during period Null 1, read R2 starts at Null 2, R3 and Null 3 and R4 at Null 4.

[0054] A complete set of the signals discussed above and defining the DRAM shared memory interface are shown in the timing diagram of FIGURE 4.

[0055] FIGURE 5 is a conceptual diagram of a switching system 500 utilizing shared memory 100 according to the inventive concepts. In this case, switching system is depicted as having ten dataports, although in actual applications, the number of ports can be substantially larger. For simplicity, the transmission media routers and other telecommunications circuitry connecting the end user to the switch system 100 are not shown. In this example, the system includes digital telephones 501, digital network 502, a workstation 503, personal computers (PCs) 504, fax machine 505, video teleconferencing equipment 506 and a digital private branch exchange (PBX) 507. Switching system 100 is under the control of switch controls 508, which may be a microprocessor or controller dedicated to switching system 100, or may be a processor or computer controlling a much larger telecommunications system of which system 500 is only a small part.

[0056] As discussed above, switching system 100 advantageously uses shared-memory to connect any two ports together. For example, digital telephone 501a on port 1 can be connected to digital telephone 501b on port 9 through shared-memory. Similarly, video teleconferencing equipment 506a and 506b, respectively on ports 5 and 10, can similarly be connected through shared-memory according to the present inventive principles. Shared memory as described in this invention, applies to any memory shared (for data access) by at least two processors, controllers or their chip sets. Shared memory with at least one data port can also be a single chip solution (SOC - System on Chip) where logic is embedded with at least one processor/controller e.g., single chip cell phone solution. Shared memory can also be a 'memory dominant IC' in a SIC (System In Chip) solution.

[0057] FIGURE 6 is a high level block diagram of an alternate embodiment of shared-memory switch 100 utilizing independent Load (Read) and Store (Send) data paths between bus 104 and shared memory 102.

[0058] FIGURES 7 through 11 illustrate how the above inventions can be applied for better system bus utilization and turnaround time. Address, command, control and data are strobed on both edges of the system clock (raising and falling edges of clock). Although the invention has been described with reference to specific embodiments, these descriptions are not meant to be construed in a limiting sense. Various modifications of the disclosed embodiments, as well as alternative embodiments of the invention will become apparent to persons skilled in the art upon reference to the description of the invention. It is therefore contemplated that the claims will cover any such modifications or embodiment that fall within the true scope of the invention. The address bus and data bus can operate at the same frequency, or, different frequencies. The address and data buses can be unidirectional as well as bidirectional. Pre-fetched addresses can be programmed into a mode-register, so that, 'long pages' (page = 1 row of data) can store multiple packets. Various burst lengths are possible. Various word lengths, predetermined or programmable on-the-fly, are also possible. The invention can be used as a stand alone memory, a system-in-chip (a module of logic, memory, mixed signal IC's), a system-on-chip (logic, memory embedded on one IC) or any combination there of.

[0059] Although the invention has been described with reference to a specific embodiments, these descriptions are not meant to be construed in a limiting sense. Various modifications of the disclosed embodiments, as well as alternative embodiments of the invention will become apparent to persons skilled in the art upon reference to the description of the invention. It should be appreciated by those skilled in the art that the conception and the specific embodiment disclosed may be readily utilized as a basis for modifying or designing other structures for carrying out the same purposes of the present invention. It should also be realized by those skilled in the art that such equivalent constructions do not depart from the spirit and scope of the invention as set forth in the appended claims.

ATTORNEY DOCKET NO.
17200- P043US

21

PATENT

[0060] It is therefore, contemplated that the claims will cover any such modifications or embodiments that fall within the true scope of the invention.